

## REMARKS

Claims 1 – 20 are pending. By this amendment, the preambles of claims 1 and 12 are amended. The amendments to the preambles of claims 1 and 12 clarify the function of the claimed invention. No new matter is introduced. Applicants respectfully request reconsideration and issuance of a Notice of Allowance.

Applicants thank Examiner Pwu for the courtesies extended to Applicants' representative during a March 25, 2005 personal interview. The substance of the interview is incorporated in the remarks that follow.

On page 2 the Office Action rejects claims 1 – 20 under 35 U.S.C. 102(e) over U.S. Patent 6,438,705 to *Chao et al.* (hereafter *Chao*). This rejection is respectfully traversed.

### **Claim 1:**

The Office Action asserts that *Chao* discloses each feature of claim 1, and cites various portions of *Chao* to support its assertion. Applicants have carefully reviewed *Chao* and strongly disagree with the Examiner's comparison of *Chao* and claim 1, and the Examiner's corresponding assertions.

*Chao* is directed to a cluster management system in which Cluster Services track performance of nodes and resources within the cluster, and restructure the cluster in the event of a failure of one or more resources or nodes. *Chao* designates this process as resource failover. See column 6, lines 24 – 31. *Chao* improves over prior art systems by allowing the failover to occur between any two nodes in a large cluster of nodes. When a computer system within the cluster is restarted, any resources that are on-line within another node of the cluster will be prevented from restarting. See column 5, lines 32 – 35. This restart prevention is provided by the addition of "hidden resources" within a resource group that check the state of all other resources within that resource group and prevent restart of a running resource. See column 8, lines 21 – 27.

*Chao* also describes, in the Background of the Invention section of the specification, a prior art system, Microsoft Cluster Services (MSCS) that is designed to provide high availability for a computer cluster, including resource failover in a two-node computer cluster. When a resource fails, MSCS will either restart the resource on the local node or move the resource group to the other node, depending on the resource restart policy and the resource group failover policy and cluster status. See column 3, lines 23 – 39.

One element of MSCS is a “special resource” called a quorum resource. *Chao* notes that there is one quorum resource in every cluster. The quorum resource stores data durably, is accessible by ether node in the cluster, and can be forcibly acquired by one node to the exclusion of the other node. *Chao* states that the quorum resource is effectively a “global control lock” for the cluster such that the node that has acquired the quorum resource “uniquely defines the cluster.” See column 4, lines 9 – 11. As *Chao* further notes, the purpose of the “global control lock” is to prevent partitioning of the cluster such that the second node must always join with the node that has acquired the quorum resource, thereby preventing a partitioned cluster with all its attendant problems. See column 4, lines 11 – 17. In essence, the quorum resource is a configuration file that lists all the resources that will exist in a cluster. The node that has acquired the quorum resource can communicate with and use the resources listed in the quorum resource, and no other node may do so until that node is part of the same cluster. See column 4, lines 18 – 40.

*Chao* extends the prior art concepts of the quorum resource by invoking a database that can contain configuration and status information for all the resources and resource groups (resources may belong to a unique resource group), and that meets all the requirements of the prior art quorum resource. See column 9, lines 40 – 49. *Chao*’s system uses the database to maintain a multi-cluster arrangement of services that can accommodate resource failure, addition of resources, and addition of nodes, and that can control cluster partitioning. *Chao* also invokes a “cluster quorum condition” that prevents a resource group from being brought online unless three quorum conditions are met:

- 1) the partition has majority quorum, i.e., more than half of all nodes defined in the cluster configuration database has joined the cluster and is in that partition, or
- (2) the partition has exactly half of the nodes as defined in the cluster configuration database and no other partitions of the same size exist, or
- (3) the partition has exactly half of the nodes as defined in the cluster configuration database while another partition contains the other half of the nodes and the node with the lowest ID value is in the former partition.

See column 13, lines 19 – 46.

However, *Chao* does not disclose or suggest:

communicating a request by a First Node to establish a lock on a Resource accessible through the Cluster; determining whether the at least one Peer Node on the Cluster holds an active lock on the Resource; if an active lock on the Resource is not held by any of the at least one Peer Node, approving the lock request; and if an active lock on the Resource is held by any of the at least one Peer Node further comprising:

determining for each active lock held on the Resource whether the requested lock conflicts with the active lock; if the requested lock does not conflict with the active lock, approving the lock request; and if the requested lock conflicts with the active lock, denying the lock request,

as recited in claim 1.

In particular, and as discussed during the personal interview, *Chao* does not disclose “communicating a request by a First Node to establish a lock on a Resource accessible through the Cluster.” Although the Office Action asserts that *Chao* does disclose this element of claim 1, the Examiner cites no part of *Chao* to support the assertion. In fact, there is nothing in *Chao* that discloses or suggest that a node of a cluster communicates any type of lock request. The only time the term “lock” is even used in *Chao* is the mere mention of a “global control lock” at column 4, line 9. The mere existence of the word “lock” in *Chao* cannot support this asserted anticipation. In the context used in *Chao*, a node establishes this “global control lock” by “seizing the quorum resources.” But *Chao* does not disclose, explicitly or inherently, how this “seizure” occurs, nor does *Chao* suggest that any communications between the nodes occurs in connection with the “seizure.” Obviously the Examiner did not provide a citation to support his contention because nothing in *Chao* discloses or suggest the claim element of “communicating a request by a First Node to establish a lock on a Resource accessible through the Cluster.” For this reason alone, claim 1 is patentable.

As also discussed during the personal interview, *Chao* does not disclose or suggest “determining whether the at least one Peer Node on the Cluster holds an active lock on the Resource,” as recited in claim 1. Again, *Chao* merely notes that a first node seizes the quorum resource, thereby establishing the global control lock, and that the second node must then join with the first node if the second node is to be part of the cluster containing the first node. See column 4, lines 10 – 13. For this additional reason, claim 1 is also patentable.

Since *Chao* does not disclose or suggest either of the first two elements of claim 1, *Chao* cannot disclose or suggest any of claim 1’s remaining elements, which each depend on the completion of the first two elements. In short, *Chao* does not disclose or suggest even one element or feature of claim 1. Accordingly, claim 1 is patentable.

#### **Claims 12 and 18:**

The remaining independent claims are each either method claims or Beauregard claims that are narrower in scope than independent claim 1, and that each contain elements,

similar to those in claim 1, that are not taught or suggested by *Chao*. Considering claim 12, *Chao* does not disclose or suggest installing a lock broker daemon on each Node of a Cluster and does not disclose or suggest establishing a Lock Broker Table associated with each lock broker daemon. Considering claim 18, *Chao* does not disclose or suggest comparing a lock name for a requested lock against a lock name of an active lock. Accordingly, claims 12 and 18 are also patentable.

**Dependent Claims:**

The dependent claims each depend from one of patentable claims 1, 12, or 18, and for this reason and the additional features they recite, the dependent claims are also patentable.

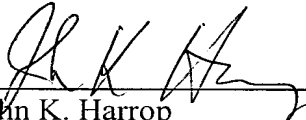
For all the reasons stated above, claims 1 – 20 are patentable. Withdrawal of the rejection of claims 1 – 20 under 35 U.S.C. § 102(e) is respectfully requested.

In view of the above remarks, Applicants respectfully submit that the application is in condition for allowance. Prompt examination and allowance are respectfully requested.

Should the Examiner believe that anything further is desired in order to place the application in even better condition for allowance, the Examiner is invited to contact Applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

Date: March 25, 2005

  
\_\_\_\_\_  
John K. Harrop  
Registration No. 41,817  
**Andrews Kurth LLP**  
1701 Pennsylvania Ave, N.W.  
Suite 300  
Washington, DC 20006  
Tel. (202) 662-3050  
Fax (202) 662-2739